

## Claims

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is as follows:

1. A method for performing dynamic parsing of structured documents, the method comprising:

obtaining a first structured document;

identifying a document type for said first structured document;

finding an extension component to process the first structured document; and

invoking the extension component upon the first structured document to generate a usable in-memory data structure.

2. The method as recited in Claim 1 where the structured document is written in XML.

15 3. A method for generating heterogeneous data structures, the method comprising:

having a first program and a second program, said first program having a set of structured data in a first data structure;

20 sending a first document including data from said first data structure to said second program; and

employing dynamic parsing to parse said first document into a second data structure for use by the second program.

4. The method as in claim 1 used for e-commerce.

5. The method as in claim 1, where the step of sending includes obtaining a request and responding to said request.

6. The method as in claim 1, where the step of parsing is performed by the second program.

7. A method for linking heterogeneous data structures, the method comprising:

10 providing a first program with a first set of data in a first data structure;

providing a second program with the first set of data in a second data structure;

15 receiving information indicating a change in one of the first and the second data structures to a third data structure; and

modifying the one of the first and the second data structure into a fourth data structure in correspondence with said third data structure.

20 8. The method as in Claim 7, wherein the step of receiving information

includes the first program sending the information.

9. A method for linking heterogeneous data structures, the method comprising:

providing a first program with a first set of data in a first data structure;

providing a second program with the first set of data in a second data structure;

5 receiving information at said second program indicating a change in the first data structure to a third data structure; and

said second program modifying the second data structure into a fourth data structure in correspondence with said third data structure.

10. The method as in Claim 9, wherein the step of receiving information comprises the first program sending the information.

11. An architecture neutral system for building clients that access a legacy system, the neutral system comprising:

15 an association module for associating each data object in the legacy system with a unique ID and with a location in a virtual table data structure;

a receiving module for receiving a request from a client for the legacy system to provide information about a requested property of a specific data object  
20 identified by the unique ID;

an identifier module for identifying the requested property by a name;

a program module for providing a query handler extension for each property which a client can request;

a query handler extension module for producing a value representing said location in said virtual table data structure; and

5 a transmitting module for transmitting the value to the client using a communication system, whereby the client can access said object in said virtual table.

12. The system as recited in Claim 11, wherein the query handler extension module includes a program executable on the identified object.

10 13. A method for building clients that access a legacy system, the method comprising:

associating each data object in the legacy system with a unique ID;

15 making a request from a client system for the legacy system to provide information about a requested property of a specific data object identified by the unique ID;

identifying the requested property by a name;

providing a query handler extension for each property which a client can request;

20 the query handler extension producing a value representing the location of said data object in a virtual table data structure; and

transmitting the value to the client for client queries regarding said data object.

14. The method as recited in Claim 13, wherein the query handler extension comprises a program executable on the identified object.

15. An architecture neutral system for building clients that  
5 access a legacy system, the neutral system comprising:

at least one client location;

at least one server location;

10 at least one client and server for transmitting two kinds of messages, a first message being a synchronous query/response, and a second message being an asynchronous subscription based event notification, whereby arbitrary data structures can be rendered into a standard communication format by applying said the contents of an asynchronous subscription based event notification for providing synchronous query/response communications.

15

16. The system as recited in Claim 15, wherein the synchronous query/response is used to submit queries from the client to the server.

20 17. The system as recited in Claim 15, wherein a client request includes an ID and a property.

18. The system as recited in Claim 17, wherein the communication link locates an object associated with the ID, and invokes an extension registered to handle queries for the property.

19. The system as recited in Claim 15, wherein the extension returns a structured text document to be transmitted to the client as a response.

20. The system as recited in Claim 19, wherein the structured text document is in XML format.  
5

21. A method for building clients that access a legacy system, the method comprising:

forming a simple bi-directional communication link between each of the clients and a server; and

10 transmitting along said communication link two kinds of messages, a first message being a synchronous query/response, and a second message being an asynchronous subscription based event notification, to allow arbitrary data structures to be rendered into a standard communication format.

15 22. The method as recited in claim 21, further comprising employing synchronous event notification based on a subscription based communication model.

23. The method as recited in Claim 22, further comprising the client initiating an event.

20 24. The method as recited in Claim 22, further comprising the server initiating an event.

25. The method as recited in Claim 23, wherein an event includes an event type and event-specific information describing the event.

26. The method as recited in Claim 21, further comprising components on the server registering listeners to receive notifications of particular event types.

5 27. The method as recited in Claim 21, further comprising components on the client registering listeners to receive notifications of particular event types.

28. The method as recited in Claim 21, further comprising providing an application that accesses legacy systems and data.

10 29. The method as recited in Claim 21, further comprising adding low-tech "extensions" to provide services.

30. The method as recited in Claim 21, further comprising providing access both to legacy data and to legacy code.

15 31. A method of creating replicas in a computing environment comprising at least a first and a second machine, the method comprising:

moving data from a first machine to a second machine to form said replica; and

20 communicating an updating event on the first machine to the second machine to keep the data consistent, said updating event being associated with said data.

32. The method as recited in Claim 31, wherein said replica includes only a portion of said data.

25 33. The method as recited in Claim 31, wherein said data is in a first form on said first machine, and said replica is transformed into a second form on said second machine.

34. The method as recited in Claim 31, further comprising communicating an updating event on the second machine to the first machine to keep the data consistent.

35. The method as recited in Claim 31, wherein the step of  
5 moving includes responding to at least one query.

36. An apparatus for dynamically parsing structured documents,  
the apparatus comprising:

a plurality of parsers, each parser for parsing a particular structured document type;

10 a registry for associating each of said parsers with a corresponding document type;

a recognizor to recognize the document type of each structured document; and

15 an invoker, which for each structured document invokes one of said parsers appropriate to parse said structured document.

37. A method for developing an interactive application, the method comprising:

implementing a server program containing at least code for parsing a first data structure;

20 implementing dynamic parsing on a first data structure to form a second data structure;

linking said first and second data structures; and

providing a client application which employs the processes of linking and parsing.

38. A method as recited in Claim 21, wherein the step of forming includes building a heterogeneous linked structure.

5 39. A computing architecture for providing replicated data structures comprising:

a server comprising an event management component having at least one event handler, a query management component having at least one query handler and at least one server database location  
10 for storing server data; and

a client comprising a query generator, an event management component having at least one event handler, a user interface, and a dynamic parsing component with at least one parser extension for accessing said server data.